|  |
| --- |
| 猜你喜欢是怎么实现的？  推荐系统必须要有庞大的数据作为支撑，但是我们在产品启动的时候，没有大量的用户数据，这时候就会遇到冷启动问题，解决冷启动问题，既然没有用户的数据，那我们就可以利用商品本身的数据来推荐，比如同价格，同品牌产品  基于内容的推荐：  给商品打上标签：食品，运动，衣服等等越详细越好。记录用户的行为数据，什么是行为数据？就是用户搜索了什么类别的物品，看了什么价位的，分享了什么物品，等等。当这个数据越多，我们的能做到事情就更多，这个行为是多样的，所以需要加权算法来计算出用户的偏好程度。  现在的推荐系统中，协同过滤算法（SF）非常常用，协同过滤的目标是基于用户对物品的历史评价信息，向目标用户推荐其未购买的物品。协同过滤算法可分为基于物品的，基于用户的和基于矩阵分解的。 |

|  |
| --- |
| 怎么发现热点数据？  我们主要是依赖前面的导购页面，首页，搜索页面，商品详情，购物车等。  构建一个异步系统，它可以收集交易链路上各个环节中的中间件产品的热点Key，如Nginx，缓存，RPC服务框架等这些中间件（一些中间件产品本身已经有热点统计模块）。 |

|  |
| --- |
| 电商系统中的购物车流程  SKU—定义为保存库存控制的最小可用单位，SKU就是那一款商品。  用户点击加入购物车🡪查看用户是否登陆🡪如果已登陆🡪查看redis中是否有商品信息🡪如果有，将redis的商品数据和新加入的商品数据合并🡪将合并后的商品SKU和购买数量信息添加到数据库🡪清空redis中的商品数据🡪修改购物车图标右上角的被加入的商品数量🡪添加成功。  用户没有登陆的话，加入购物车要跳转到登陆页面，把那个商品信息保存到redis里面，所以一登陆要判断redis有没有商品，有的话要自动加入，不能再选一次，提高用户体验度，没有商品的话直接加入到数据库。    用户打开购物车流程：  打开购物车流程：每次打开购物车判断是否登陆是必须的，然后还得查询物品的状态，状态如果是下架的要灰掉，并且无法勾选，库存也得重新获取。  用redis全面代替表操作，提供性能。 |

|  |
| --- |
| 购物车结算之前的订单显示的逻辑，要在购物车里面写，因为要访问redis，查看库存够不够，这个操作没有必要访问数据库，因为有的人只是凑单，查看优惠等等，而且每次进来都得查。  订单传参问题：为什么要传一个集合，两个对象， 是为了查询快，每个人知道的数据不一样，发快递的只想知道地址，不用链表查询等 集合的话用几个实体类，里面放一个list泛型是具体的对象即可。  怎么解决第二个商品库存不够了，让第一个也回滚，自旋的时候不重复买，  Index控制次数，自旋的时候-1， |

|  |
| --- |
| 猜你喜欢模块大体是大数据系统分析，java查询显示，是根据用户的收藏，分享，购买记录，购物车记录以及接下来要过的节日等。协同过滤算法：用户的协同过滤算法：基于用户的协同过滤算法是寻找与A用户有相似行为的所有B用户所喜欢的而A用户还不知道的物品推荐给A用户。 |

|  |
| --- |
| 只有前台的查询会保存到redis 后台的商品不会存入redis中，因为会有下架的，状态不对的商品，存入redis没有意义。 |

|  |
| --- |
| 你都用到什么表？  广告信息表：自增ID、广告位置、广告网址、广告名称、广告开始时间、广告结束时间、排序号、广告点击数、广告类型、删除标注、创建时间  广告位置表：自增ID、广告类型、有效状态、广告位置代码、排序号、建议高度、建议宽度 |